Лабораторна робота № 9. Дослідження методів узагальнення: LDA, SVD (LSA), PCA, t-SNE та ансамблевих методів: Random Forest, XGBoost, LightGBM, CatBoost, AdaBoost

Мета: сформулювати практичні навички у використанні та порівнянні методів узагальнення даних та ансамблевих методів машинного навчання, включаючи LDA, SVD (LSA), PCA, t-SNE, Random Forest, XGBoost, LightGBM, CatBoost та AdaBoost. Розвинути здатність аналізувати великі набори даних, визначати найбільш ефективні методи для конкретних завдань класифікації або редукції розмірності, а також опанувати навички налаштування гіперпараметрів для покращення результатів моделі.

Теоретичні відомості

Методи узагальнення даних, також відомі як методи редукції розмірності, використовуються для зменшення кількості вхідних змінних у наборі даних, шукаючи новий набір ознак, який зберігає найважливішу інформацію з оригінального набору. Ці методи дозволяють спростити моделі, зменшити обчислювальні витрати та поліпшити інтерпретацію даних, зберігаючи при цьому якомога більше корисної інформації.

Приклади методів узагальнення:

1. **PCA (Principal Component Analysis)** – метод лінійної редукції розмірності, який використовується для зменшення кількості змінних у датасеті, зберігаючи при цьому якомога більше інформації. PCA шукає новий, менш розмірний простір для представлення даних.
2. **t-SNE (t-Distributed Stochastic Neighbor Embedding)** – нелінійний метод зниження розмірності, особливо ефективний для візуалізації великих наборів високорозмірних даних.
3. **LDA (Linear Discriminant Analysis)** – метод, що використовується для знаходження лінійних комбінацій ознак, які найкраще розділяють два або більше класів об’єктів або подій.
4. **SVD (Singular Value Decomposition)** та **LSA (Latent Semantic Analysis)** – **SVD** є математичним методом, який дозволяє розкладати матрицю на три інші матриці, а **LSA** використовує **SVD** для аналізу відносин між набором документів та термінами, що містяться в них, для виявлення семантичних

структур.

Ансамблеві методи в машинному навчанні – це підходи, що комбінують прогнози з кількох моделей навчання для покращення загальної точності, надійності та ефективності моделі. Ансамблеві методи базуються на принципі, що група «слабких» моделей, які працюють разом, може перевершити одну «сильну» модель.

Приклади ансамблевих методів:

- **Random Forest** – ансамблевий метод, що будує багато дерев рішень під час тренування та виводить клас, що є модою класів (класифікація) або середнє прогнозоване деревами (регресія).
- **XGBoost (Extreme Gradient Boosting)** – оптимізований алгоритм градієнтного бустингу, що використовується для збільшення швидкості та ефективності обчислень.
- **LightGBM** – швидкий, ефективний алгоритм градієнтного бустингу, який використовує алгоритми розбиття на основі гістограм для обробки великих обсягів даних.
- **CatBoost** – алгоритм, який автоматично обробляє категорійні змінні та забезпечує високу точність прогнозування.
- **AdaBoost (Adaptive Boosting)** – алгоритм, що послідовно виправляє помилки попередніх класифікаторів та надає їм ваги для формування сильного класифікатора.

Приклад виконання роботи

Завдання

1. Оберіть та завантажте датасет відповідно до свого варіанту. Завантажте ваш набір даних як тимчасовий файл у середовище Google Colab та ініціалізуйте його.
2. Проаналізуйте датасет та визначте ознаки, які будуть використовуватися у методах узагальнення та ансамблевих методах. Як цільову зміну класифікації використовуйте клас заданий у таблиці.
3. Виконайте очищення даних: видаліть або заповніть пропущені значення, опрацюйте викиди якщо це потребується.
4. Використайте алгоритми узагальнення: 1) PCA; 2) t-SNE; 3) LDA; 4) SVD (LSA) та ансамблеві методи: 1) Random Forest; 2) XGBoost; 3) LightGBM; 4) CatBoost; 5) AdaBoost. Виконайте оцінку та візуалізуйте результати для кожної моделі.
5. Виконайте класифікацію за допомогою різних ансамблевих моделей для довільних вхідних даних відповідно до вашої цільової змінної.
6. Проаналізуйте отримані результати, обґрунтуйте вибір найкращої моделі для даного набору даних.
7. За бажанням проекспериментуйте з гіперпараметрами для моделей задля покращення результатів.

Контрольне запитання №1

Вкажіть результуюче значення Accuracy для Random Forest.
Відповідь: _____

Контрольне запитання №2

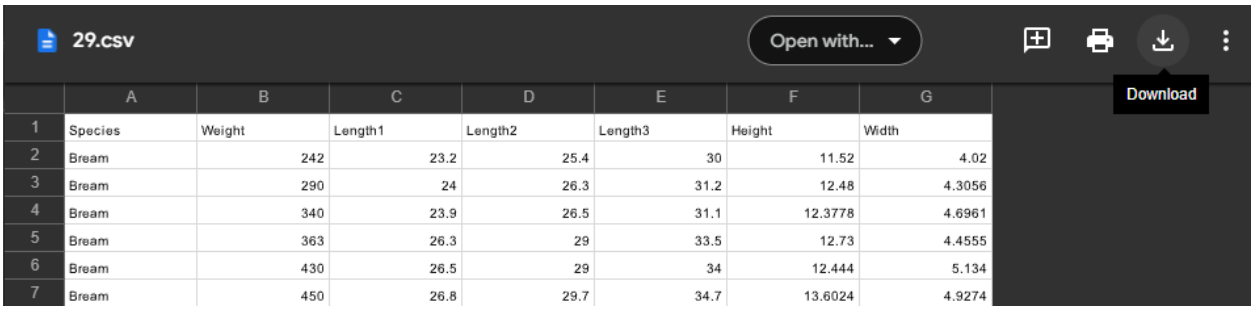
Вкажіть результуюче значення Accuracy для CatBoost.
Відповідь: _____

Контрольне запитання №3

Вкажіть результуюче значення Accuracy для LightGBM.
Відповідь: _____

Хід роботи

1. Збережемо та імпортуємо набір даних за варіантом.



	A	B	C	D	E	F	G
1	Species	Weight	Length1	Length2	Length3	Height	Width
2	Bream	242	23.2	25.4	30	11.52	4.02
3	Bream	290	24	26.3	31.2	12.48	4.3056
4	Bream	340	23.9	26.5	31.1	12.3778	4.6961
5	Bream	363	26.3	29	33.5	12.73	4.4555
6	Bream	430	26.5	29	34	12.444	5.134
7	Bream	450	26.8	29.7	34.7	13.6024	4.9274

Рисунок 8.1. Датасет на Google Drive

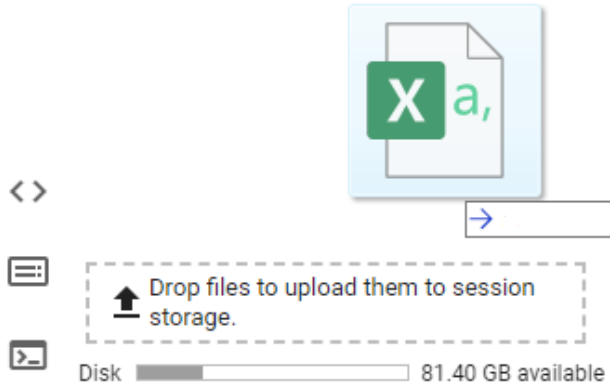


Рисунок 8.2. Завантаження датасету до Google Colab

2. Встановлення необхідних бібліотек та модулів

```
import warnings
warnings.filterwarnings('ignore')

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA, TruncatedSVD
from sklearn.manifold import TSNE
from sklearn.ensemble import RandomForestClassifier, AdaBoostClassifier
from xgboost import XGBClassifier
from sklearn.metrics import accuracy_score, confusion_matrix
import ipywidgets as widgets
from IPython.display import display, clear_output
from sklearn.preprocessing import LabelEncoder
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis as
LDA
```

```
from lightgbm import LGBMClassifier
from catboost import CatBoostClassifier
```

3. Ініціалізація набору даних в Google Colab.

```
# Завдання №1. # Завантаження датасету
data = pd.read_csv("/content/29.csv")
data.head()
```

	Species	Weight	Length1	Length2	Length3	Height	Width
0	Bream	242.0	23.2	25.4	30.0	11.5200	4.0200
1	Bream	290.0	24.0	26.3	31.2	12.4800	4.3056
2	Bream	340.0	23.9	26.5	31.1	12.3778	4.6961
3	Bream	363.0	26.3	29.0	33.5	12.7300	4.4555
4	Bream	430.0	26.5	29.0	34.0	12.4440	5.1340

Рисунок 8.3. Перші 5 рядків набору даних

4. Визначимо цільову змінну, як Species, ознаками будуть: Weight, Length1, Length2, Length3, Height, Width.

5. Виконаємо очищення даних

```
# Перевірка на відсутні значення
print(data.isnull().sum())
```

```
Species    0
Weight     0
Length1    0
Length2    0
Length3    0
Height     0
Width      0
dtype: int64
```

Рисунок 8.4. Пошук нульових значень у датасеті

Порожні значення відсутні, тож видалення або заповнення виконувати не треба.

```
# Видалення або заповнення відсутніх значень
# data.dropna(inplace=True) # Видалення

# data.fillna(data.mean(), inplace=True) # Заповнення середніми значеннями
```

Виконаємо візуалізацію однієї з ознак набору даних, а саме «Length2»

```
# Візуалізація викидів
plt.figure(figsize=(10, 6))
sns.boxplot(x=data['Length2'])
plt.title('Boxplot before removing outliers')
plt.show()
```

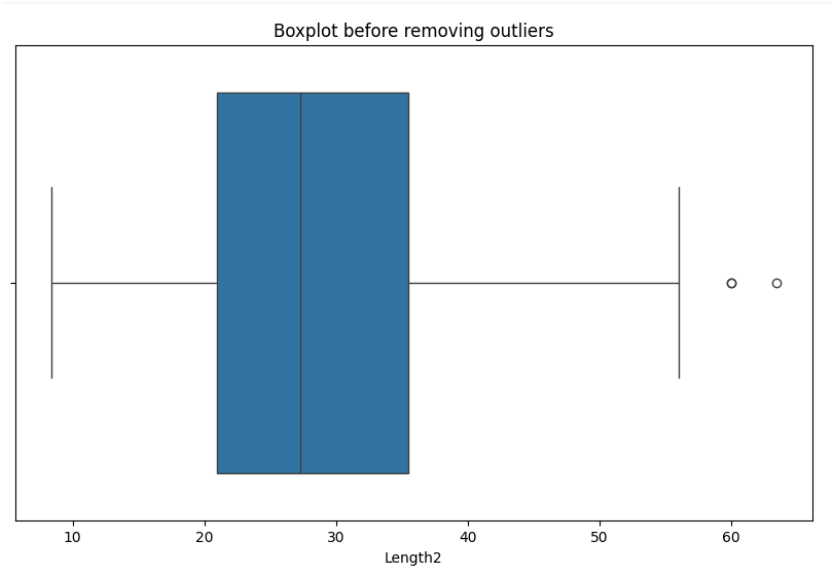


Рисунок 8.5. Діаграма викидів для Length2 перед очищенням

Виконаємо очищення викидів для «Length2».

```
# Видалення викидів
feature_processing = 'Length2'
q_low = data[feature_processing].quantile(0.01)
q_hi  = data[feature_processing].quantile(0.99)
data = data[(data[feature_processing] < q_hi) & (data[feature_processing]
> q_low)]
```

```
plt.figure(figsize=(10, 6))
sns.boxplot(x=data['Length2'])
plt.title('Boxplot after removing outliers')
plt.show()
```

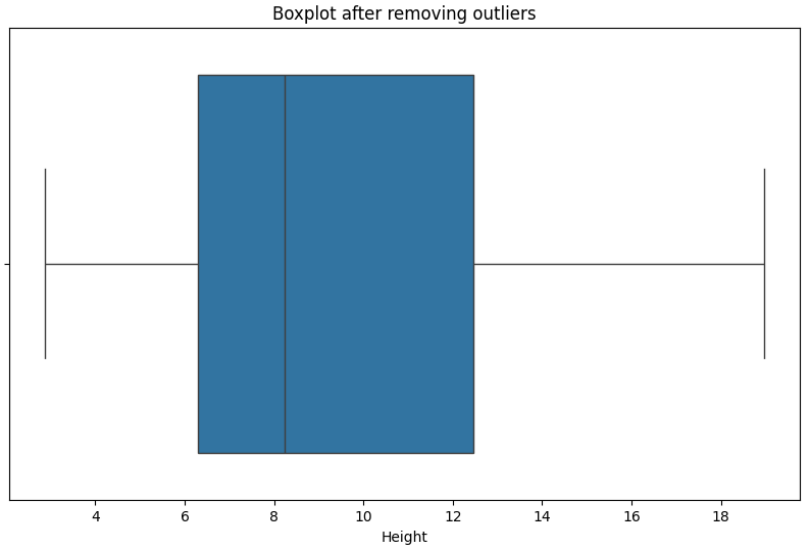


Рисунок 8.6. Діаграма викидів для Length2 після очищення

Подібні операції виконаємо для інших ознак.

6. Розподілимо дані на тренувальні та тестові набори з урахуванням ознак та цільової змінної. В тестовому наборі даних видалимо стовпець «Species», що класифікується та виконаємо розподіл тестових та тренувальних наборів даних. Водночас виконаємо перетворення міток класів у числовий формат за допомогою LabelEncoding.

```
X = data.drop('Species', axis=1)
y = data['Species']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,
random_state=42)

label_encoder = LabelEncoder()
y_train_encoded = label_encoder.fit_transform(y_train)
y_test_encoded = label_encoder.transform(y_test)
```

Виконаємо нормалізацію даних.

```
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

7. Підготуємо та використаємо алгоритми узагальнення PCA, t-SNE, LDA та SVD (LSA), а також ансамблеві методи Random Forest, XGBoost, LightGBM, CatBoost та Adaboost.

```
model_widget = widgets.Dropdown(
    options=['PCA', 't-SNE', 'LDA', 'SVD (LSA)', 'Random Forest',
'XGBoost', 'LightGBM', 'CatBoost', 'AdaBoost'],
    description='Model:',
```

```
)

# Віджети для налаштування гіперпараметрів
hyperparameter_widgets = {
    'n_components': widgets.IntSlider(min=2, max=5, value=2,
description='n_components:'),
    'max_depth': widgets.IntSlider(min=1, max=10, value=3,
description='Max Depth:'),
    'learning_rate': widgets.FloatSlider(min=0.01, max=0.3, value=0.1,
step=0.01, description='Learning Rate:'),
    'n_estimators': widgets.IntSlider(min=10, max=200, value=100,
description='N Estimators:'),
    'silent': widgets.Checkbox(value=True, description='Silent:'),
}

hyperparameters_ui = widgets.VBox([])

def update_hyperparameters_ui(*args):
    model_name = model_widget.value
    children = [model_widget]
    if model_name in ['Random Forest', 'XGBoost', 'LightGBM', 'CatBoost',
'AdaBoost']:
        children.append(widgets.IntSlider(min=10, max=200, value=100,
description='n_estimators:'))
        children.append(widgets.IntSlider(min=1, max=10, value=3,
description='max_depth:'))
    if model_name in ['XGBoost', 'LightGBM', 'CatBoost']:
        children.append(widgets.FloatSlider(min=0.01, max=0.3, value=0.1,
step=0.01, description='learning_rate:'))
    if model_name == 'CatBoost':
        children.append(widgets.Checkbox(value=False,
description='Silent:'))
    hyperparameters_ui.children = children

model_widget.observe(update_hyperparameters_ui, 'value')

output = widgets.Output()

def train_and_visualize(button):
    with output:
        output.clear_output(wait=True)
        model_name = model_widget.value
        n_components = hyperparameter_widgets['n_components'].value
        max_depth = hyperparameter_widgets['max_depth'].value
        learning_rate = hyperparameter_widgets['learning_rate'].value
        n_estimators = hyperparameter_widgets['n_estimators'].value
        silent = hyperparameter_widgets['silent'].value

        global y_train, y_test
```

```
        if model_name in ['PCA', 't-SNE', 'LDA', 'SVD (LSA)']:
            if model_name == 'PCA':
                model = PCA(n_components=n_components)
            elif model_name == 't-SNE':
                model = TSNE(n_components=n_components)
            elif model_name == 'LDA':
                model = LDA(n_components=n_components)
            elif model_name == 'SVD (LSA)':
                model = TruncatedSVD(n_components=n_components)

            X_transformed = model.fit_transform(X_train_scaled, y_train)
            plt.figure(figsize=(10, 6))
            plt.scatter(X_transformed[:, 0], X_transformed[:, 1],
c=y_train_encoded)
            plt.title(f'{model_name} visualization')
            plt.colorbar()
            plt.show()
        else:
            if model_name == 'Random Forest':
                model = RandomForestClassifier(n_estimators=n_estimators,
max_depth=max_depth)
            elif model_name == 'XGBoost':
                model = XGBClassifier(n_estimators=n_estimators,
max_depth=max_depth, learning_rate=learning_rate)
            elif model_name == 'LightGBM':
                model = LGBMClassifier(verbose= -1,
n_estimators=n_estimators, max_depth=max_depth,
learning_rate=learning_rate)
            elif model_name == 'CatBoost':
                model = CatBoostClassifier(n_estimators=n_estimators,
max_depth=max_depth, learning_rate=learning_rate, silent=silent)
            elif model_name == 'AdaBoost':
                model = AdaBoostClassifier(n_estimators=n_estimators)

            model.fit(X_train_scaled, y_train_encoded)
            y_pred_encoded = model.predict(X_test_scaled)
            accuracy = accuracy_score(y_test_encoded, y_pred_encoded)
            cm = confusion_matrix(y_test_encoded, y_pred_encoded)
            print(f"Model: {model_name}, Accuracy: {accuracy:.4f}")
            sns.heatmap(cm, annot=True, fmt="d")
            plt.title(f'Confusion Matrix: {model_name}')
            plt.show()

train_button = widgets.Button(description="Train and Visualize")
train_button.on_click(train_and_visualize)

hyperparameter_ui = widgets.VBox([v for k, v in
hyperparameter_widgets.items()])
ui = widgets.VBox([model_widget, hyperparameter_ui, train_button, output])
```

```
display(ui)
```

8. Виконаємо тестування алгоритмів узагальнення та ансамблевих методів

Використаємо алгоритм узагальнення PCA з показником «n_components» = 2.

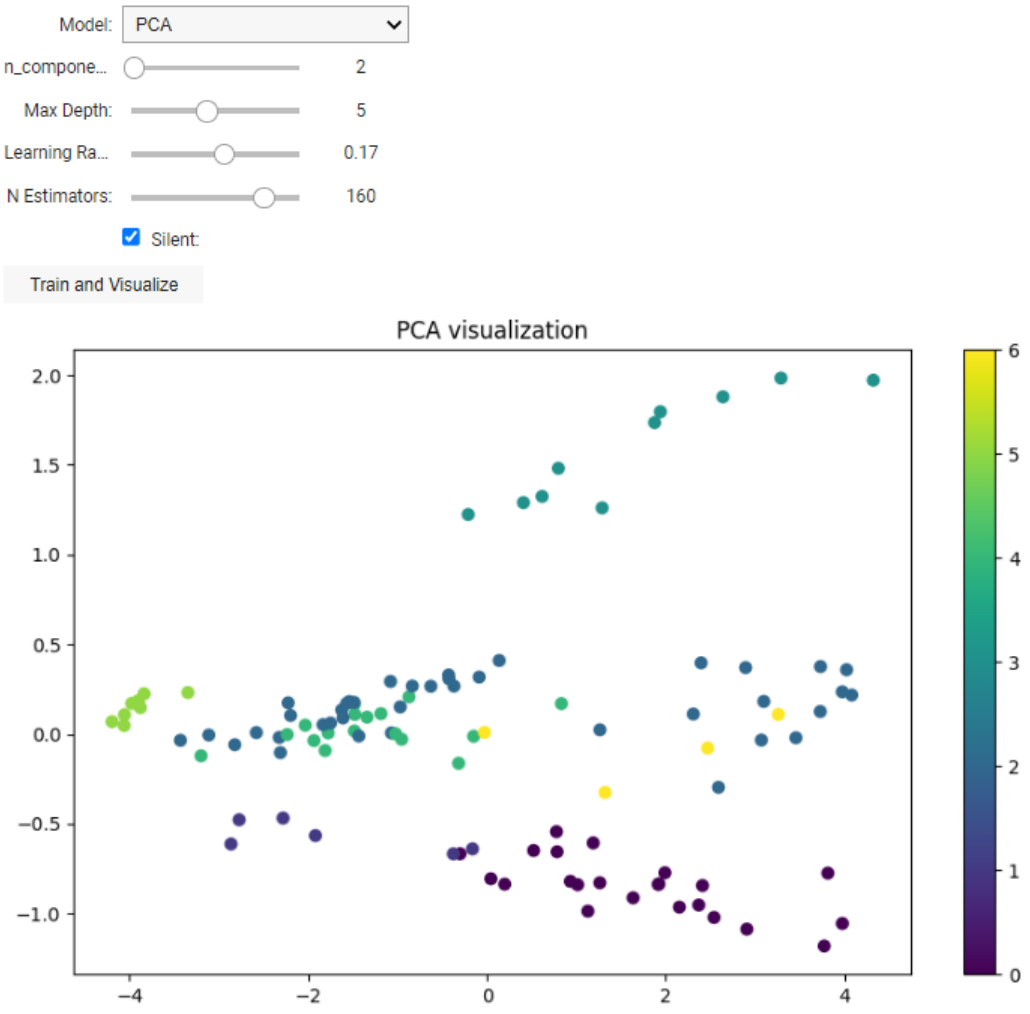


Рисунок 8.7. Результат використання моделі PCA

Використаємо алгоритм узагальнення t-SNE з показником «n_components» = 2.

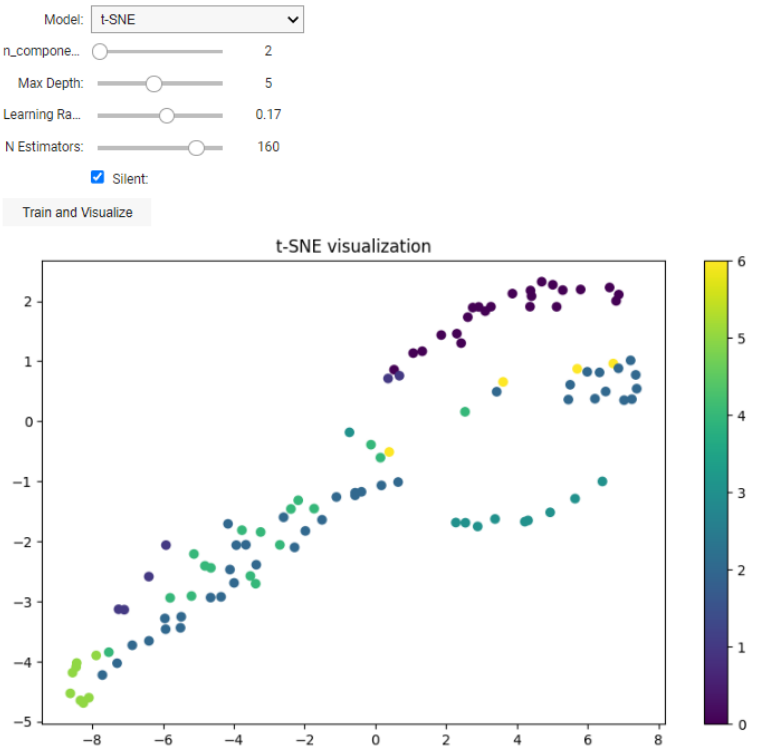


Рисунок 8.8. Результат використання моделі t-SNE

Використаємо алгоритм узагальнення LDA з показником «n_components» = 2.

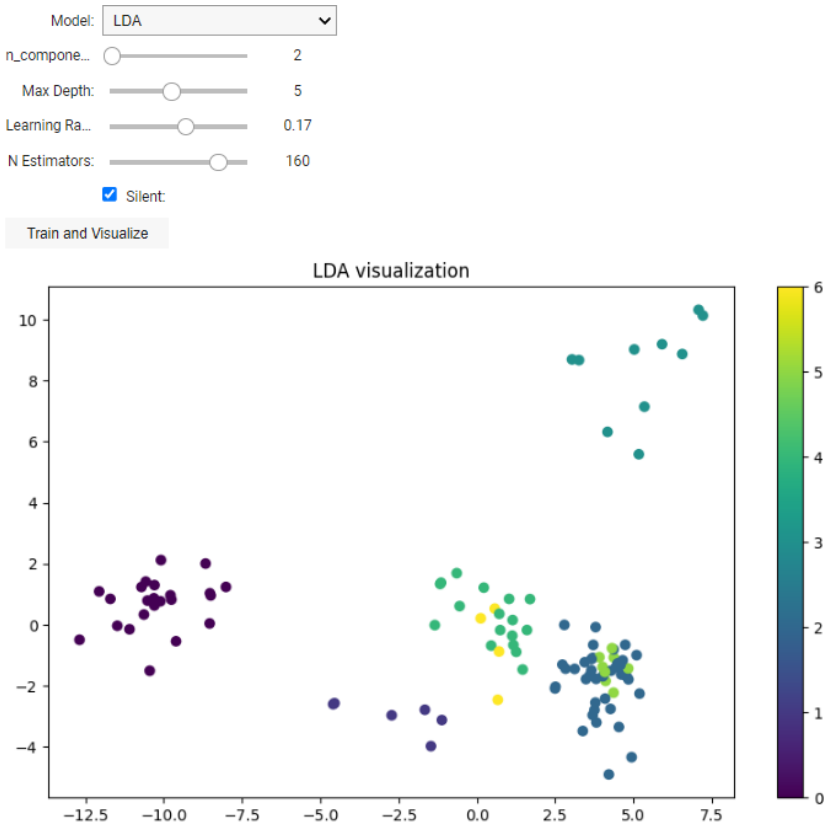


Рисунок 8.9. Результат використання моделі LDA

Використаємо алгоритм узагальнення SVD (LSA) з показником «n_components» = 2.

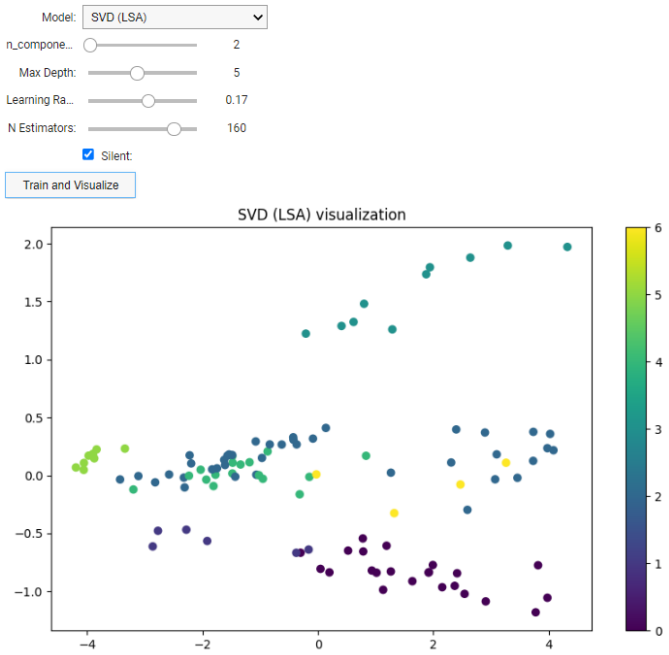


Рисунок 8.10. Результат використання моделі SVD (LSA)

Використаємо ансамблевий метод Random Forest з показниками «n_estimators» = 160, «max_depth» = 5.

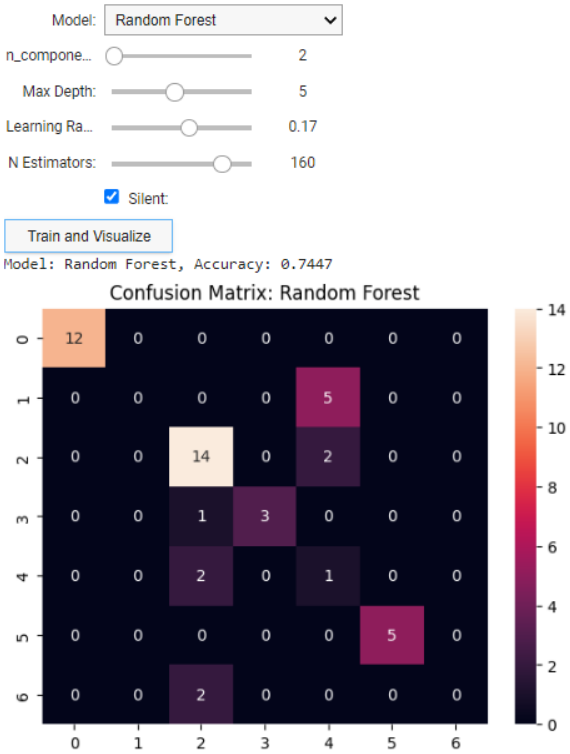


Рисунок 8.11. Результат використання ансамблевої моделі Random Forest

Використаємо ансамблевий метод XGBoost з показниками «n_estimators» = 160, «max_depth» = 5, «learning_rate» = 0.17.

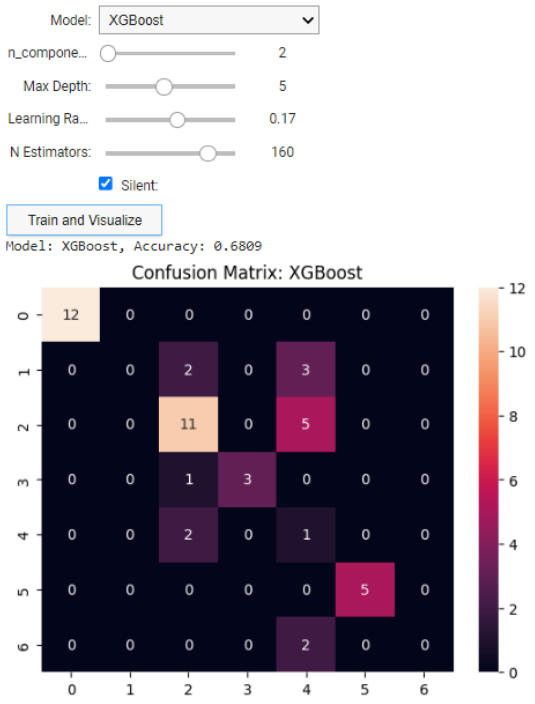


Рисунок 8.12. Результат використання ансамблевої моделі GXBoost

Використаємо ансамблевий метод LightGBM з показниками «n_estimators» = 160, «max_depth» = 5, «learning_rate» = 0.17.

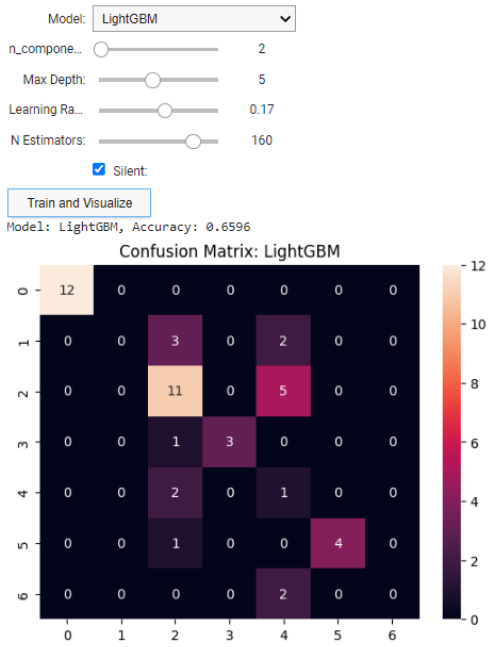


Рисунок 8.13. Результат використання ансамблевої моделі LightGBM

Використаємо ансамблевий метод CatBoost з показниками «n_estimators» = 160, «max_depth» = 5, «learning_rate» = 0.17, «silent» = True.

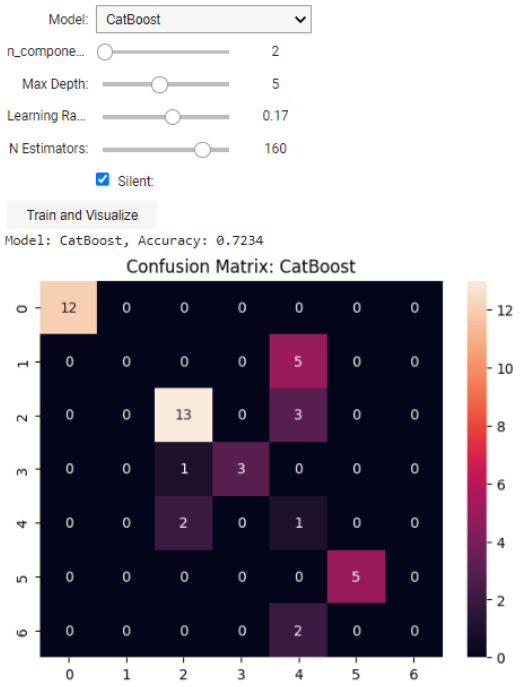


Рисунок 8.14. Результат використання ансамблевої моделі CatBoost

Використаємо ансамблевий метод AdaBoost з показниками «n_estimators» = 160.

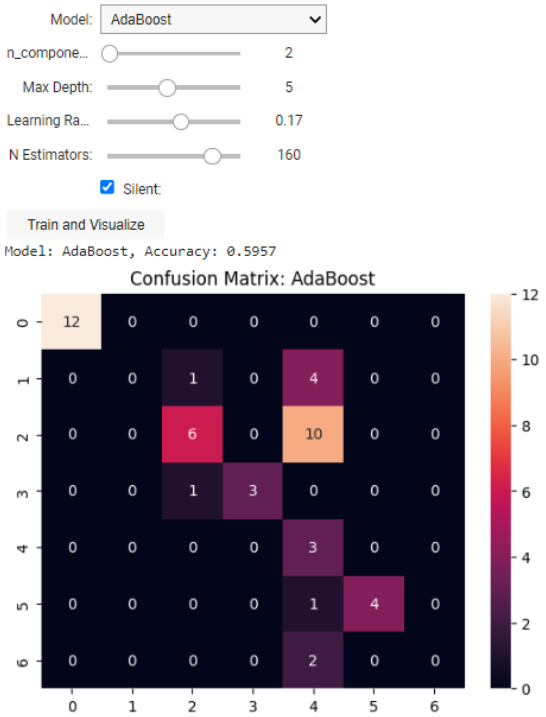


Рисунок 8.15. Результат використання ансамблевої моделі AdaBoost

9. Виконаємо прогнозування на довільних даних

```
model_widget = widgets.Dropdown(
    options=['Random Forest', 'XGBoost', 'LightGBM', 'CatBoost',
'AdaBoost'],
    description='Model:',
)

n_estimators_widget = widgets.IntSlider(min=10, max=200, value=100,
description='n_estimators:')
max_depth_widget = widgets.IntSlider(min=1, max=10, value=3,
description='max_depth:')
learning_rate_widget = widgets.FloatSlider(min=0.01, max=0.3, value=0.1,
step=0.01, description='learning_rate:')


hyperparameter_widgets = widgets.VBox([n_estimators_widget,
max_depth_widget, learning_rate_widget])

input_widgets = [widgets.FloatText(description=f'Feature {i+1}:') for i in
range(X_train.shape[1])]

classify_button = widgets.Button(description="Classify")
output = widgets.Output()

def classify(b):
    with output:
        clear_output()
        data_to_classify = [w.value for w in input_widgets]
        model = prepare_model(
            model_choice=model_widget.value,
            n_estimators=n_estimators_widget.value,
            max_depth=max_depth_widget.value,
            learning_rate=learning_rate_widget.value,
        )
        prediction = model.predict([data_to_classify])
        print(f"Predicted class: {prediction[0]}")
        y_pred = model.predict(X_test)
        accuracy = accuracy_score(y_test, y_pred)
        print(f"Accuracy: {accuracy:.4f}")

classify_button.on_click(classify)

def prepare_model(model_choice, n_estimators, max_depth, learning_rate):
    if model_choice == 'Random Forest':
        model = RandomForestClassifier(n_estimators=n_estimators,
max_depth=max_depth)
    elif model_choice == 'XGBoost':
        model = XGBClassifier(n_estimators=n_estimators,
max_depth=max_depth, learning_rate=learning_rate)
```

```
    elif model_choice == 'LightGBM':
        model = LGBMClassifier(verbose= -1, n_estimators=n_estimators,
max_depth=max_depth, learning_rate=learning_rate)
    elif model_choice == 'CatBoost':
        model = CatBoostClassifier(n_estimators=n_estimators,
depth=max_depth, learning_rate=learning_rate, silent=True)
    elif model_choice == 'AdaBoost':
        model = AdaBoostClassifier(n_estimators=n_estimators,
learning_rate=learning_rate)
    model.fit(X_train, y_train)
    return model

display(widgets.VBox([model_widget, hyperparameter_widgets] +
input_widgets + [classify_button, output]))
```

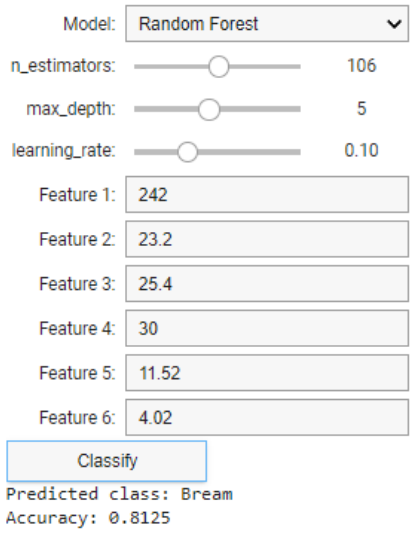


Рисунок 8.16. Результати прогнозування для моделі Random Forest

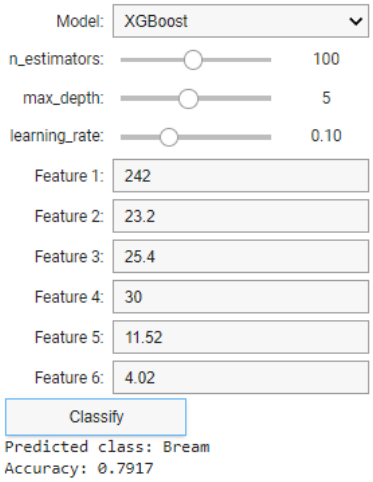


Рисунок 8.17. Результати прогнозування для моделі XGBoost

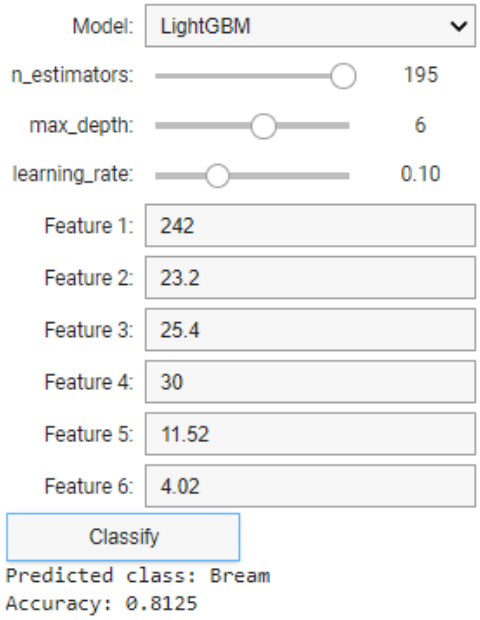


Рисунок 8.18. Результати прогнозування для моделі LightGBM

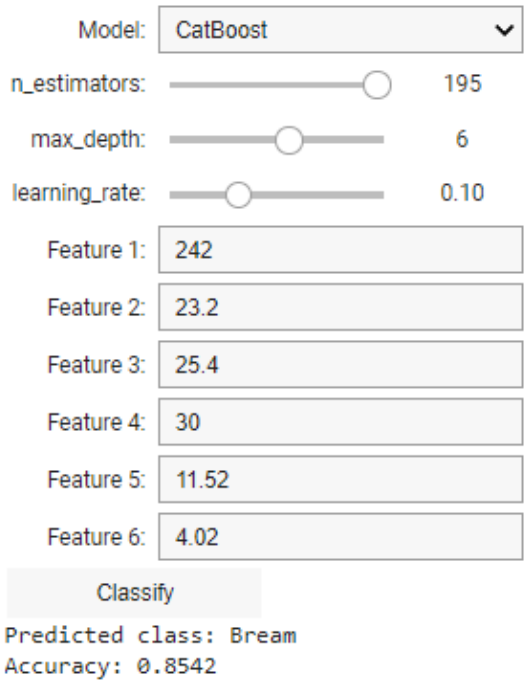


Рисунок 8.19. Результати прогнозування для моделі CatBoost

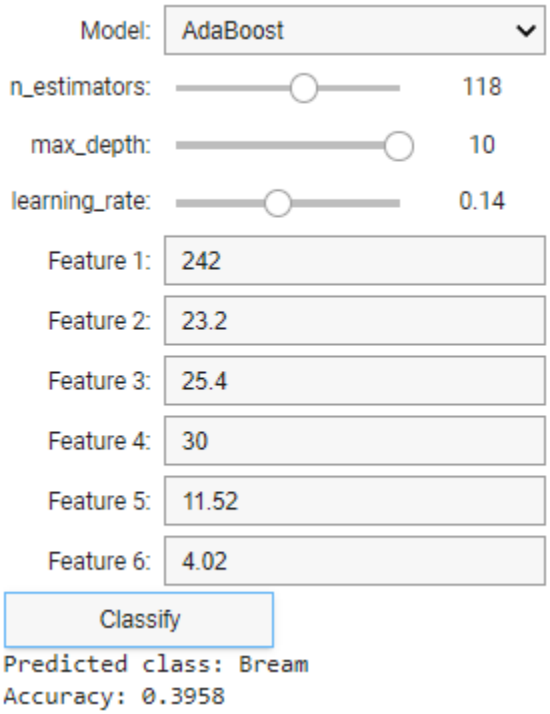


Рисунок 8.20. Результати прогнозування для моделі AdaBoost

Таким чином, …

Контрольне запитання №1

Вкажіть результуюче значення Accuracy для Random Forest.
Відповідь: **0.7447**

Контрольне запитання №2

Вкажіть результуюче значення Accuracy для CatBoost.
Відповідь: **0.7234**

Контрольне запитання №3

Вкажіть результуюче значення Accuracy для LightGBM.
Відповідь: **0.6596**

Висновки: …

Завдання для виконання лабораторної роботи

1. Оберіть та завантажте датасет відповідно до свого варіанту. Завантажте ваш набір даних як тимчасовий файл у середовище Google Colab та ініціалізуйте його.
2. Проаналізуйте датасет та визначте ознаки, які будуть використовуватися у методах узагальнення та ансамблевих методах. Як цільову зміну класифікації використовуйте клас заданий у таблиці.
3. Виконайте очищення даних: видаліть або заповніть пропущені значення, опрацюйте викиди якщо це потребується.
4. Використайте алгоритми узагальнення: 1) PCA; 2) t-SNE; 3) LDA; 4) SVD (LSA) та ансамблеві методи: 1) Random Forest; 2) XGBoost; 3) LightGBM; 4) CatBoost; 5) AdaBoost. Виконайте оцінку та візуалізуйте результати для кожної моделі.
5. Виконайте класифікацію за допомогою різних ансамблевих моделей для довільних вхідних даних відповідно до вашої цільової змінної.
6. Проаналізуйте отримані результати, обґрунтуйте вибір найкращої моделі для даного набору даних.
7. За бажанням проекспериментуйте з гіперпараметрами для моделей задля покращення результатів.

	Посилання на датасети				
№	**1**	**5**	**9**	**13**	**17**
Classification	Brand	label	Microcalcification	category	wifi
№	**2**	**6**	**10**	**14**	**18**
Classification	class	Type	diabetes	R	is_safe
№	**3**	**7**	**11**	**15**	**19**
Classification	POP	Species	Car	class	quality
№	**4**	**8**	**12**	**16**	**20**
Classification	ocean_proximity	CarName	poutcome	custcat	Species

Контрольне запитання №1

Вкажіть результуюче значення Accuracy для Random Forest.
Відповідь: _____

Контрольне запитання №2

Вкажіть результуюче значення Accuracy для CatBoost.
Відповідь: _____

Контрольне запитання №3

Вкажіть результуюче значення Accuracy для LightGBM.

Відповідь: _____

Welcome to Google Colab

https://colab.research.google.com/drive/1spYcnLtT7DdbErQLG7PsTZHftc02Dpq7?usp=sharing